TrueCity: Real and Simulated Urban Data for Cross-Domain 3D Scene Understanding

Duc Nguyen[*1] , Yan-Ling Lai[*1] , Qilin Zhang[1] , Prabin Gyawali[1] , Benedikt Schwab[1] , Olaf Wysocki[1,2] , Thomas H. Kolbe[1]

[1] Technical University of Munich, [2] CV4DT, University of Cambridge

(duc.nguyen, ... thomas.kolbe)@tum.de ; okw24@cam.ac.uk

* equal contribution

Abstract

3D semantic scene understanding remains a long-standing challenge in the 3D computer vision community. One of the key issues pertains to limited real-world annotated data to facilitate generalizable models. The common practice to tackle this issue is to simulate new data. Although synthetic datasets offer scalability and perfect labels, their designer-crafted scenes fail to capture real-world complexity and sensor noise, resulting in a synthetic-to-real domain gap. Moreover, no benchmark provides synchronized real and simulated point clouds for segmentation-oriented domain shift analysis. We introduce TrueCity, the first urban semantic segmentation benchmark with cm-accurate annotated real-world point clouds, semantic 3D city models, and annotated simulated point clouds representing the same city. TrueCity proposes segmentation classes aligned with international 3D city modeling standards, enabling consistent evaluation of synthetic-to-real gap. Our extensive experiments on common baselines quantify domain shift and highlight strategies for exploiting synthetic data to enhance real-world 3D scene understanding. We are convinced that the TrueCity dataset will foster further development of sim-to-real gap quantification and enable generalizable data-driven models. The data, code, and 3D models are available online: https://github.com/tum-gis/TrueCity.

1. Introduction

Semantic segmentation of point clouds remains a critical yet unsolved task in 3D computer vision. One of the main impediments in this area is the limited availability of high-quality, semantically annotated real-world 3D data. To address this issue, many researchers have turned to synthetic data generation by simulating sensory inputs in virtual environments [8, 9, 21, 41, 44]. These synthetic datasets offer an appealing alternative due to their scalability, cost-effectiveness, and precise ground truth annotations.

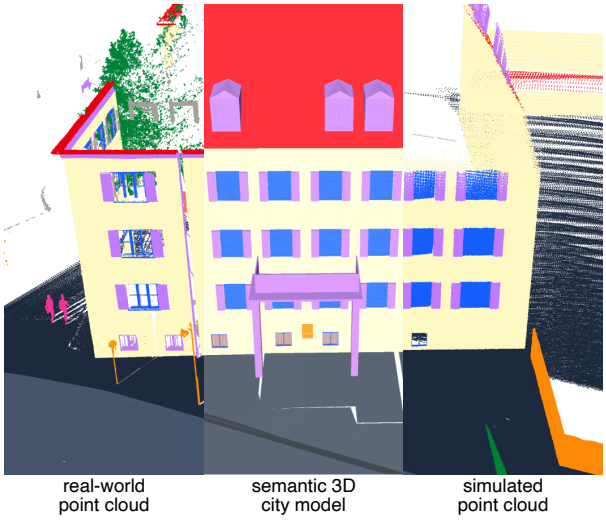

Figure 1. TrueCity introduces real-world annotated point clouds, a semantic 3D city model, and 3D-model simulated point clouds of the same location, enabling coherent evaluation of the sim-to-real domain gap in 3D scene understanding.

However, a significant drawback of using simulated data lies in the large domain shift between synthetic and real-world point clouds. This is especially apparent when dealing with urban scenes which are highly challenging scenarios owing to changing scanning distances, various object material properties, and many dynamic events. Simulated environments often rely on fictitious and designer-crafted 3D scenes that fail to capture the full complexity, variability, and noise present in real-world settings [9, 41, 44]. Another

limitation is the lack of benchmark datasets comprising synchronized real-world 3D point clouds and 3D environment paired with their simulated counterparts. Lack of such data hampers comprehensive quantification of the domain shift between synthetic and real domains. This research gap not only impedes the development of robust segmentation models but also limits our understanding of how synthetic data can best be leveraged to improve real-world performance.

Another general challenge in developing urban segmentation methods is the high heterogeneity and definition of urban classes. This variability often leads to misinterpretation of object characteristics and hinders the creation of large-scale, high-diversity datasets. In particular, there is often a taxonomic and geometric mismatch between the segmented urban elements and the internationally recognized and standardized modeling classes. The issue is frequently overlooked, yet it poses a significant obstacle to analyze per class sim-to-real shift, especially apparent in ray-penetrable glass objects which simulation fails to capture [8, 57].

To address these challenges, our TrueCity contributes in:

- Proposing novel urban semantic segmentation classes derived from international modeling standards;
- Realizing the new class design on the introduced TrueCity benchmark dataset comprising real-world cm-accurate labeled point clouds, its derived semantic 3D city model structured according to the CityGML 2.0 standard, and simulated point clouds representing the same city (Fig. 1);
- Analyzing domain shift between synthetic and real point clouds on the segmentation task enabled by data synchronization.

2. Related Work

2.1. Domain Shift in Point Cloud Segmentation

Inherently, machine and deep learning methods require a large amount of training data, which is expensive to collect. Especially in the context of semantic segmentation of 3D point clouds, the scarcity of well-annotated data for each sensor type has become a notable challenge [12, 45, 60]. The immediate application of a model trained on one sensor type to another is often infeasible owing to the different laser scanning patterns, point distribution, and point cloud size. For instance, Shen et al. [42] show semantic segmentation decreases its performance significantly when trained on terrestrial laser scanning (TLS) and inferred on mobile laser scanning (MLS) point cloud, i.e., 76.5% to 32.0% in their experiments. Recent research endeavors have shifted towards integrating domain adaptation techniques accounting for the sensor differences and showing promising results [60], e.g., improving segmentation accuracy by up to 13% [42]. Yet, the performance is still insufficiently robust, showing large variations across datasets and reaching only around 56% IoU on challenging datasets [52].

Furthermore, unlike in 2D image domain [35], there is a lack of large real-world point clouds to train a generic classifier that can be adapted for multiple scenarios [59]. Consequently, recent years have witnessed a surge in methods investigating the adoption of simulated point clouds complementing real-world scenarios [8, 54, 60]. Current simulators often represent fictitious scenes designed manually [9, 41, 53]. This fact poses yet another challenge of not only accounting for the sensor type differences but also the domain gap between the simulated and real-world data. As reflected by experiments conducted by Xiao et al. [60], the performance for combining simulated and real data can oscillate in the range of 55-65% accuracy depending on the scenario and benchmark dataset, underscoring the importance of further domain-gap investigations.

Although recently researchers have shown that weakly-supervised [26, 50] and self-supervised [61, 63] methods show promising results, there are still necessitating validation sets and their performance can be limited, e.g., can reach only up to 60% when deployed for object detection, as shown in the review of Zeng et al. [61].

Worth noting is also the parallel submission to a journal [2]. There, the focus lies on providing the workflow to generate simulated data and provide deterministic and stochastic tools to validate the simulation. Also, unlike in this TrueCity submission, no synchronized dataset is introduced; TrueCity further harmonizes the semantic classes with the international standards; and importantly TrueCity provides revised test, validation, and training split without mixing real with synthetic point clouds in local neighborhoods, thus enabling consistent domain shift evaluation in separate spatial regions.

2.2. Urban Point Cloud Segmentation Datasets

A wide range of point cloud semantic segmentation benchmarks have been introduced in recent years (Tab. 1). However, the availability of such datasets remains much smaller that of 2D image segmentation benchmarks, both in terms of the number of datasets and the volume of annotated instances. Street-level urban point cloud segmentation datasets represent particularly challenging scenarios, where even transformer-based methods show limited performance, e.g., Point Transformer achieves only around 42% IoU on the facade segmentation task in the ZAHA dataset [59].

Despite progress, none of the current benchmarks provide synchronized real and simulated point clouds captured at the same geographic location (except aerial data which is out of scope for this publication [5]). Notable datasets such as DELIVER [24] and Paris-CARLA-3D [8] include simulation data, but the synthetic scenes are not representing the corresponding real sites. In Paris-CARLA-3D, Paris scans are paired with CARLA-generated fictitious towns, and the underlying 3D real city assets are unavailable, precluding

scene-level re-simulation and making systematic analysis of domain shifts, and thus a fair estimate of the domain gap difficult.

Another challenge lies in the high variation of class definitions across urban segmentation datasets, which hinders unified, consistent, and fair comparisons of algorithmic performance: The minimum number is eight and the maximum is 50 in the analyzed datasets (Tab. 1). The lack of standardized class representations also poses difficulties for developing robust transfer learning approaches. In related fields such as geomatics, architecture, and civil engineering, international standardization bodies have long established urban object taxonomies for modeling [1, 10, 59]. These standardized class definitions, however, remain largely overlooked in the design of point cloud segmentation datasets, which limits the direct applicability of their outputs to standardized semantic modeling workflows. Here, the notable exceptions are ZAHA [59], TUM-FAÇADE [56], and ArCH [29], grounding their classes in international standards. Yet, they focus on either facade-only classes (ZAHA, TUM-FAÇADE), excluding any other urban classes; or on cultural-heritage-specific classes (ArCH), limits their application to generic urban scenarios. Worth noting are also works focusing on generating synthetic images from virtual 3D assets, yet they are out of scope for the 3D point cloud benchmarks [11, 31].

2.3. Standardized Semantic City Models

For the representation of cities and landscapes in terms of their semantics, geometry, topology, and appearance, the international standard CityGML has become widely adopted by municipalities and entire countries [17, 58]. CityGML version 2.0 was released by the Open Geospatial Consortium (OGC) in 2012 and defines a conceptual data model, which is based on the standards from the ISO 191XX series of geographic information standards and therefore supported by geographic information systems. The standard specifies concepts and class definitions for representing buildings, bridges, tunnels, vegetation, city furniture, and transportation infrastructure across different level of details (LoDs) [17]. Combined with buildings, detailed facades, vegetation, and city furniture, the street space can be represented in a comprehensive, semantic, and interoperable manner. As of 2024, there are approximately 216.5 million building models available worldwide as open CityGML datasets [58]. This includes the building stocks in LoD 2 of Germany, Switzerland, Poland, and large parts of Japan, which are provided and maintained by public authorities with stable object identifiers. Moreover, road networks and roadside objects can also be modeled in the OpenDRIVE standard for high-definition mapping, traffic and driving simulation applications [1, 37]. Objects defined in OpenDRIVE using parametric geometries can be transformed into CityGML's explicit geometries through a dedicated conversion method [39].

3. TrueCity Benchmark Dataset

3.1. 3D Semantic Road Space Classes

As shown in Table 1, there is a scarcity of point cloud benchmark datasets capturing the same environment through both real and simulated laser scanning. Addressing this scarcity requires an environment model with semantic information to enable the derivation of semantically labeled point clouds from laser scanning simulation. To ensure compatibility with established standardized data models, we propose a class list of 12 classes harmonized with the standards CityGML 2.0 [17] and OpenDRIVE 1.4 [1]. Leveraging standardized class definitions facilitates seamless integration and reuse in downstream methods and applications, such as 3D road space and facade semantic reconstruction [15, 57]. A detailed description of these classes is provided in Table 2, along with their corresponding classes defined in the standards. The standards further specify type categories, functions, and usage lists for objects; for details see [1, 17].

3.2. Real-World Data Acquisition

For the physical environment of the TrueCity benchmark dataset, we selected the inner city of Ingolstadt, a mid-sized German town with a dense and challenging urban setting. Figure 2 shows the dataset covers four streets totaling about 500 m, featuring adjacent facades of 2-4-story buildings, vegetation, street furniture, vehicles, and pedestrians.

Real-World Laser Scanning The high-precision laser scans are conducted by the company 3D Mapping Solutions using their Mobile-Road-Mapping-System (MoSES), which is mounted on a minivan to collect the real-world point clouds [19, 43]. In total, 113 million points are recorded, with densities of up to 3,000 pts/m^2, while the setup achieves a relative accuracy of 1-3 cm. The inertial measurement unit is complemented by odometer sensors and a high-precision differential GPS with real-time kinematic (RTK) correction data from the German satellite positioning service (SAPOS), ensuring accurate georeferencing [19]. The georeferenced real-world point clouds are provided in the projected coordinate reference system (CRS) UTM Zone 32N (EPSG:25832).

Semantic Annotation Our labeling of real-world point clouds follows a three-step process. First, we divide the point cloud into parts of connected components based on their spatial distance. Second, the ground surfaces, including roads and sidewalks, are separated from elevated objects using the cloth simulation filtering (CSF) algorithm [62]. Third, we manually select the corresponding subset of points and assign it a class ID from Table 2. Finally,

Name	Year	Sensor	Acquisition?	# Classes	Real city?	3D models?	Standardized classes?	Sim and Real synced?
Oakland 3D [30]	2009	MLS	real	44	✓	✗	✗	-
Sydney Urban Objects Dataset [7]	2013	MLS	real	26	✓	✗	✗	-
Paris-rue-Madame database [40]	2014	MLS	real	27	✓	✗	✗	-
iQumulus [49]	2015	MLS	real	8	✓	✗	✗	-
TUM-MLS-2016 [65]	2016	MLS	real	9	✓	✗	✗	-
semantic3D.net [18]	2017	TLS	real	9	✓	✗	✗	-
Paris-Lille-3D [36]	2018	MLS	real	50	✓	✗	✗	-
SynthCity [16]	2019	MLS	simulated	9	✗	✗	✗	-
A2D2 [13]	2020	MLS	real	38	✓	✗	✗	-
ArCH [28]	2020	TLS/MLS/UAV	real	10	✓	✗	✓	-
Toronto-3D [46]	2020	MLS	real	8	✓	✗	✗	-
KITTI-360 [25]	2021	MLS	real	19	✓	✗	✗	-
Paris-CARLA-3D [8]	2021	MLS	real + simulated	23	~	~	✗	✗
TUM-FAÇADE [56]	2022	MLS	real	17	✓	✗	✓	-
HelixNet [27]	2022	MLS	real	9	✓	✗	✗	-
SUD [14]	2023	MLS	real	8	✓	✗	✗	-
DELIVER [24]	2025	MLS	simulated	25	✗	✓	✗	-
ZAHA [59]	2025	MLS	real	15	✓	✗	✓	-
TrueCity (ours)	2025	MLS	real + simulated	12	✓	✓	✓	✓

Table 1. Point cloud benchmark datasets for urban semantic segmentation.

#	Class	Description	Corresponding standard class	
			CityGML 2.0 [17]	OpenDRIVE 1.4 [1]
1	RoadSurface	Vehicle-allowed surfaces without sidewalks	(Auxiliary)TrafficArea	LaneSectionLRLane
2	GroundSurface	Pedestrian-allowed surfaces without road surface	(Auxiliary)TrafficArea, OuterFloorSurface	LaneSectionLRLane
3	CityFurniture	Vertical urban installation without building-attached objects	CityFurniture	Signal, RoadObject
4	Vehicle	Parked or moving vehicles	–	–
5	Pedestrian	Standing or moving persons	–	–
6	WallSurface	Building parts without roofs, installations, facade elements	WallSurface	RoadObject
7	RoofSurface	Building parts forming roof structures	RoofSurface	RoadObject
8	Door	Openings allowing entering objects with gates	Door	–
9	Window	Openings and its outer blinds without entries	Window	–
10	BuildingInstallation	Building-attached installation	BuildingInstallation, OuterCeilingSurface	–
11	SolitaryVegetationObject	Vegetation with tree trunks and branches	SolitaryVegetationObject	RoadObject
12	Noise	Noisy points and any other non-annotated element	–	–

Table 2. Semantic road space classes harmonized with the class definitions of the standards CityGML 2.0 and OpenDRIVE 1.4.

all point clouds are merged, resulting in the points-per-class distribution shown in Figure 3. This phenomenon posits TrueCity in the realistic long-tail distribution regime, typical challenge of real-world data [6, 59].

3.3. Synthetic Data Acquisition

Semantic 3D City Model To achieve precise synchronization, we manually model the buildings and detailed facades using the acquired real point clouds, in accordance with the CityGML standard. The road network and roadside objects are manually curated using the same point clouds as an OpenDRIVE dataset, which we subsequently convert to CityGML 2.0 [38, 39]. This yields a comprehensive, georeferenced, and semantic model of the four streets, shown in the third row of Figure 2. We procedurally replace the coarse geometric representation of trees with detailed 3D assets and translate all 3D models into a local CRS for the laser scanning simulation, while retaining the class information.

Simulated Laser Scanning To replicate the real-world laser scanning process, we use the CARLA driving simulator, which includes a configurable LiDAR sensor model [9]. We resimulate the real-world sensor data collection route using the LiDAR parameters specified in Table 5. To further enhance realism, we incorporate a Gaussian-distributed range error proposed by [44] as follows:

$$r' = r + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \rho^2), \quad \rho = 0.02\,\mathrm{m} \tag{1}$$

The simulation accounts only for static environment objects; consequently, the synthetic point clouds exclude the *Vehicle* and *Pedestrian* classes.

4. Experiments

We evaluate on TrueCity under controlled synthetic-real (%S–%R) mixtures of 100S–0R, 75S–25R, 50S–50R, 25S–75R, and 0S–100R. We form each S–R mixture by assigning contiguous spatial segments to the synthetic or real domain, rather than interleaving points as explored in our par-

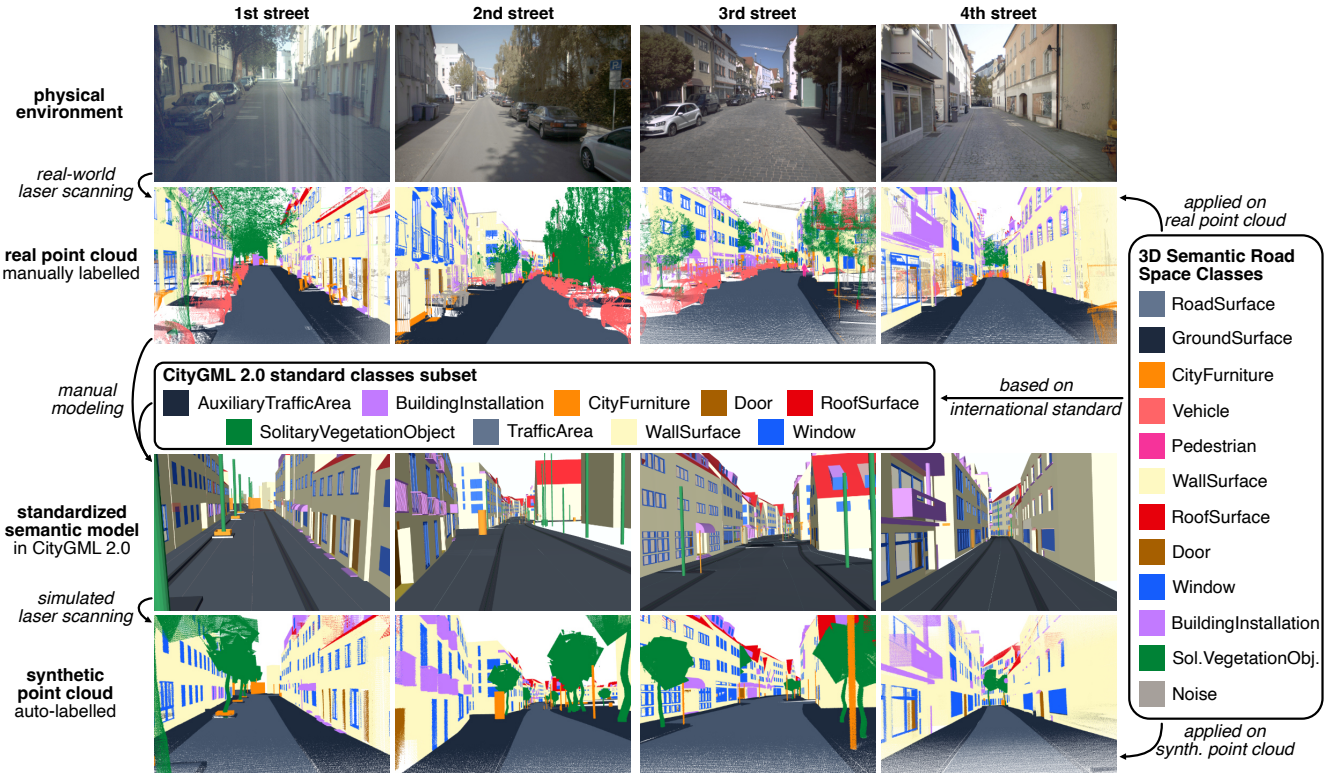

Figure 2. Real-world point cloud (2nd row), which was manually labeled according to the class list of Table 2, used for manual modeling of semantic 3D models (3rd row), which in turn were used to simulate and auto-label synthetic point clouds (4th row).

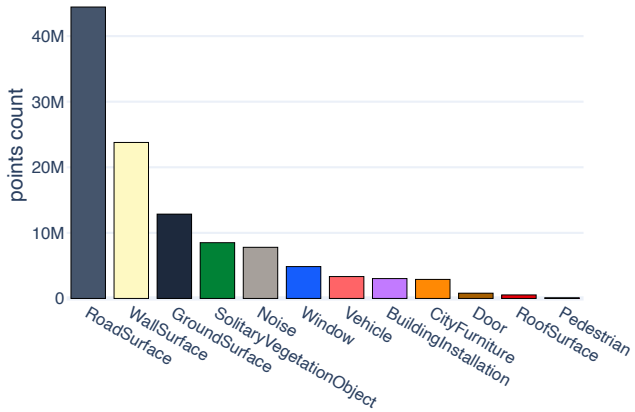

Figure 3. TrueCity represents the typical long-tail distribution challenge of real-world data.

detailed class distribution in each mixture can be found in Supplement Sec. 10.

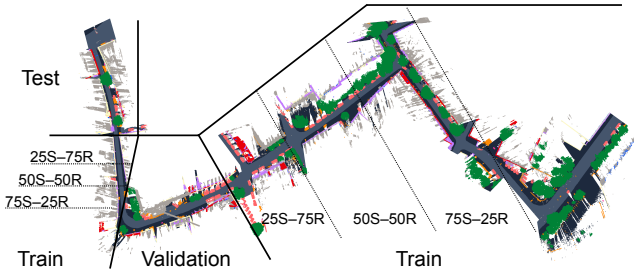

Figure 4. Top-down schematic of S–R mixtures along a continuous streetscape. Solid lines mark train/validation/test splits; dashed lines mark boundaries between contiguous synthetic and real segments for each mixture ratio.

allel submission [2]. The test and validation data remain real-only throughout the experiments. For a target real fraction r, we allocate r of the street-surface area and match the total street length across mixtures to isolate composition effects. Street length and spatial coverage are fixed across mixtures, only the raw point totals vary modestly (112.9-137.6M) due to domain-specific sampling densities. The

Evaluation protocol All results are evaluated on a fixed real-only validation/test split. Synthetic scans appear only in training when included by the mix. We report mIoU, OA, and mAcc over the 12-class TrueCity taxonomy. Aggregate results appear in Table 3. A per-class IoU breakdown for PointNet++, KPConv, and Point Transformer v1 appears in Table 4, with full results in the Supplement (Sec. 10).

4.1. Baseline Semantic Segmentation Methods

To probe synthetic-real (S–R) domain shift on TrueCity, we evaluate a representative suite of point-cloud semantic segmentation baselines widely used for urban scenes and consistently strong on established benchmarks [3, 4, 8, 46]. The suite spans three complementary families: point-based, kernel-based, and transformer-based, so that trends are not tied to a single inductive prior. See the Supplement for the detailed description (Sec. 10)

5. Results and Discussion

Synthetic data helps, but its utility depends on the model's inductive bias. Architectures with global attention handle synthetic well: Point Transformer v3 improves mIoU from 14.13% at 100S–0R to broad maximum of 25.30% at 50S–50R and 25.24% at 0S–100R. Point Transformer v1 benefits primarily from real data, moving from 16.30% at 100S–0R to 28.89% at 0S–100R, indicating a stronger reliance on sensor statistics. Hierarchical set abstraction also profits from a modest synthetic prior: PointNet++ peaks at 25.38% with 25S–75R versus 23.39% with 0S–100R, likely because synthetic coverage reduces sparsity while real data calibrates noise. Methods driven by local neighborhoods or superpoints lean heavily on real data: RandLA-Net rises from 8.98% to 17.71%, and Superpoint Transformer from 14.31% to 19.61% as the real fraction increases, consistent with their dependence on LiDAR sampling artifacts underrepresented in simulation. The practical recipe is straightforward: use synthetic as a patch, not a replacement, favor balanced mixes (around 50S–50R or 25S–75R) for strong transformers, and bias toward real data for locality-driven models while using synchronized synthetic twins to fill occlusions and thin structures.

5.1. Insignificant Domain Gap Classes

Table 4 highlights that some classes show minimal sensitivity to the proportion of real data, either maintaining strong IoU across all settings or reaching near-peak performance with limited real supervision.

WallSurface is well captured synthetically by all three models. Even without real data, Point Transformer v1 reaches 53.15% IoU and improves only modestly to 67.10% with full real supervision, a relative gain of 26.2%. KPConv shows a similar trend, attaining 73.55% with 75% real data, which corresponds to a 22.6% improvement over the synthetic-only setting (59.97%). In contrast, PointNet++ relies heavily on local sampling and interpolation. With only synthetic input, it performs poorly (0.20%), but a small fraction of real data (25%) boosts its performance dramatically to 56.90%. After this sharp correction, further gains remain limited. This confirms that large planar facades with consistent geometry can already be recognized effectively using the synthetic data. For Point Transformer v1 and KPConv, real data provides only incremental refinement, whereas PointNet++ requires minimal real supervision to overcome its synthetic-domain weakness.

RoadSurface also starts from a strong synthetic baseline: 60.90% for PointNet++ and 62.41% for Point Transformer v1. Both reach around 80% with 50% real data, corresponding to relative gains of 33.3% and 28.7%. In contrast, KPConv starts much lower at 30.64% but rises steeply to 69.26% with just 25% real data, a 126.1% improvement. This indicates that convolutional kernels rely more heavily on realistic geometrical cues, whereas PointNet++ and Point Transformer v1 are already robust under synthetic supervision. *GroundSurface* records lower absolute scores overall, but all three models show a steady upward trajectory, which again reflects moderate domain sensitivity.

SolitaryVegetationObject shows consistent trends across methods: Point Transformer v1 improves from 15.54% to 35.35% with 25% real data, KPConv rises from 45.74% to 70.51%, and PointNet++ from 0% to 50.60%. Although performance continues to increase with more real supervision, the early gains highlight that a small fraction of real data is sufficient to close much of the initial domain gap.

Overall, these categories share traits of geometric regularity and relatively simple structural context. Synthetic data provides a strong baseline, while real data mainly refines fine appearance cues. Even small proportions of real data suffice to approach optimal performance, making these classes representative of insignificant domain gap behavior.

5.2. Significant Domain Gap Classes

As shown in Table 4, several categories display substantial domain gaps, where performance depends heavily on the balance of synthetic and real supervision.

Door, *RoofSurface*, and *BuildingInstallation* remain the weakest classes across all methods, with IoU rarely exceeding 8%. Adding real data can increase scores by several orders of magnitude (e.g., Point Transformer v1 rises from 0.01% to 0.66% for *Door*), yet these gains are unstable and often collapse when real data dominates. These categories depend on fine-scale geometry and contextual cues absent from simulation, while limited real samples risk overfitting to narrow geometrical patterns.

Noise shows a more consistent improvement: PointNet++ climbs from 0% to 31.70%, KPConv from 0.31% to 34.73%, and Point Transformer v1 from 0.16% to 35.04%, corresponding to an increase of over 30% in the best case. Although Point Transformer v1 drops slightly when trained without synthetic data, all three methods display a strong positive trend, confirming that real-world variation is essential for learning this diverse class.

CityFurniture overall exhibits a U-shaped trend across all models. Performance drops sharply at intermediate real-

Model	100S–0R		75S–25R		50S–50R		25S–75R		0S–100R	
	mIoU	OA	mIoU	OA	mIoU	OA	mIoU	OA	mIoU	OA
PointNet [32]	6.03	30.36	10.74	48.10	10.89	49.29	13.10	47.99	**14.51**	**49.82**
PointNet++ [33]	9.72	34.39	20.95	62.80	23.18	**65.36**	**25.38**	63.27	23.39	63.15
RandLA-Net [20]	8.98	35.40	13.25	50.32	15.73	**59.37**	16.89	57.09	**17.71**	54.57
KPConv [48]	15.84	50.07	21.55	62.08	28.50	61.62	22.33	61.92	**29.90**	**62.80**
Point Transformer v1 [64]	16.30	57.54	19.79	60.29	23.43	67.54	24.66	**68.70**	**28.89**	67.98
Point Transformer v3 [55]	14.13	53.15	19.29	60.22	**25.30**	**65.94**	24.64	65.72	25.24	60.75
Superpoint Transformer [34]	14.31	54.17	17.01	**58.62**	14.22	54.63	19.61	**56.98**	15.96	54.64
OctFormer [51]	13.07	53.30	14.17	55.34	14.22	49.71	13.91	50.97	**17.65**	**56.28**

Table 3. TrueCity segmentation under synthetic-real (S–R) mixes. We report mIoU and OA; shifts along S–R reveal family-specific inductive biases in point-, kernel-, and transformer-based models. Bold marks the best value for each model across mixtures.

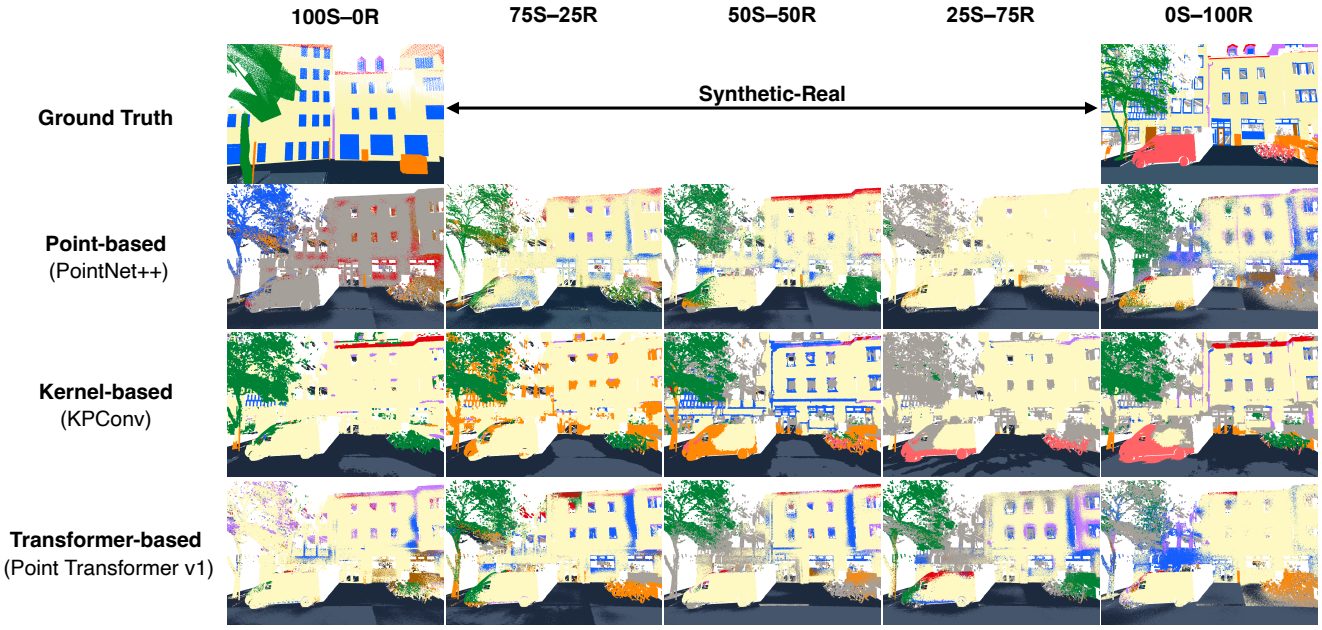

Figure 5. Qualitative impact of the synthetic–real (S–R) training mix on models from different methods (Point-based, Kernel-based and Transformer-based). We also present the ground truth synthetic and real point clouds; colors follow the TrueCity legend.

data proportions (e.g., KPConv declines from 18.25% to 1.85%), but recovers once training relies entirely on real supervision, reaching 14.57%. This suggests that mid-range ratios trigger domain conflicts, where synthetic and real samples provide competing cues, whereas fully real data resolves this inconsistency.

Window improves steadily for PointNet++ (3.61% $\rightarrow$ 14.60%) and Point Transformer v1 (3.28% $\rightarrow$ 31.29%), yet absolute scores remain modest. Their small size, thin geometry, and similarity to walls, combined with reflections and occlusions, make them difficult to segment. Synthetic data fails to capture these effects, and limited real samples cannot cover the variability, constraining performance.

Overall, for large domain gap classes, which often involve fine-scale geometry, occlusions, or high variability in appearance, finding the right synthetic-real balance is essential for stable and accurate segmentation.

5.3. Limitations and Future Work

TrueCity provides a unique combination of synchronized a) real-world cm-accurate captured point clouds, b) semantic 3D city models, and c) annotated and simulated point clouds in the same real-world location. Our comprehensive experiments were conducted on geometric values, without mixing them with any radiometric values to ensure fair comparison across three data subsets on the geometry level. Yet, the radiometry can also be included by, for example, image projection as we attach images and their trajectories taken

Class	PointNet++					KPConv					Point Transformer v1				
	100S	75S	50S	25S	0S	100S	75S	50S	25S	0S	100S	75S	50S	25S	0S
RoadSurface	60.90	73.90	**81.20**	80.30	72.50	30.64	**69.26**	54.26	56.31	54.59	62.41	61.89	**80.33**	77.93	70.44
GroundSurface	33.20	45.30	**50.40**	49.60	32.40	24.67	**36.49**	31.91	33.20	35.96	31.94	37.54	**45.54**	37.09	41.67
CityFurniture	15.20	14.60	21.30	25.50	**46.20**	**18.25**	10.09	12.67	1.85	14.57	25.89	19.50	13.51	14.23	**44.83**
Vehicle	**3.50**	0.00	0.18	0.51	0.01	0.00	0.00	38.93	**70.33**	63.64	0.00	0.51	4.41	1.51	**6.54**
Pedestrian	0.00	0.00	0.00	**0.32**	0.04	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**0.05**	0.00	0.00
WallSurface	0.20	56.90	63.50	**65.70**	63.20	59.97	63.21	68.54	**73.55**	71.03	53.15	61.42	64.39	65.32	**67.10**
RoofSurface	0.00	**0.80**	0.30	0.10	0.30	0.82	0.01	0.47	0.00	**1.79**	0.10	0.29	**0.92**	0.30	0.08
Door	0.00	0.00	0.20	0.10	**0.70**	0.00	0.00	0.00	0.00	0.00	0.01	0.13	0.51	**0.66**	0.01
Window	3.61	7.30	4.50	4.20	**14.60**	1.86	0.43	**18.21**	1.17	2.89	3.28	5.61	7.61	11.78	**31.29**
BuildingInstallation	0.00	0.70	3.00	1.70	**4.60**	**7.85**	3.02	0.85	0.85	3.72	3.08	5.24	6.69	**8.87**	3.41
Sol.VegetationObj.	0.00	**50.60**	39.30	45.60	14.40	45.74	70.51	**81.45**	0.41	77.13	15.54	35.35	34.70	43.22	**52.31**
Noise	0.00	1.30	14.30	30.90	**31.70**	0.31	5.58	**34.73**	30.26	33.43	0.16	9.98	22.43	**35.04**	28.96

Table 4. Per-class IoU (↑) for one representative per family: PointNet++ (Point-based), KPConv (Kernel-based), and Point Transformer v1 (Transformer-based) across synthetic-real (S–R) training mixes. Columns 100S, 75S, 50S, 25S, and 0S denote the synthetic fraction used in training (e.g., 100S–0R). Evaluation is on the fixed test split. Bold marks the best value for each model across mixtures. Full per-class tables appear in the Supplementary Material (Sec. 10).

during the mobile mapping acquisition. Owing to the high acquisition cost, we acknowledge that such a dataset cannot reach the scalability rate of image-based datasets, as noted in other well-established point cloud datasets, e.g., ScanNet [6]. In TrueCity the focus lies on static objects, as the real-world capture is collected at one timestamp, and introducing de-synchronized dynamic objects would impede per-class domain shift analysis. Nevertheless, since we provide all three subsets of data, dynamic objects can be simulated in various traffic scenarios.

6. Conclusion

In this paper, we present TrueCity, the benchmark dataset comprising real and simulated point clouds synchronized with underlying semantic 3D city models representing the same geographic location.

Based on our extensive experiments, we conclude that TrueCity enables in-depth coherent analysis of the domain gap across different network architectures, removing the uncertainty factor stemming from fictitious, de-synchronized scenes. Another finding of this study indicates that real point clouds can be partially replaced by simulated ones without compromising performance: several methods achieve comparable or even superior results with 50% real data compared to full real training. For example, Point Transformer v3 achieves 65.94% OA with 50% synthetic data, an 8.5% relative improvement over the all-real setting (60.75%).

Also, we observe that material complex classes suffer from simplistic assumptions in simulators (e.g., rays not penetrating through glass). For instance, without real data, *Window* IoU is only 3.28% for Point Transformer v1 and 3.61% for PointNet++, showing the limits of synthetic data. In contrast, well-represented and primitive-like classes can be simulated to a large extent (e.g. *WallSurface* reaches 59.97% IoU for KPConv trained on synthetic data only). Finally, complex non-manmade classes such as *SolitaryVegetationObject* remain difficult to synthesize owing to over-simplified representations in the underlying 3D model. Yet, even small amounts of real data bring substantial improvements, e.g., PointNet++ improves from 0.00% at 100S–0R to 50.60% at 75S–25R. We are convinced this dataset can foster further research on cross-domain gap analysis and large-scale data simulation.

Acknowledgements

We would like to thank the company 3D Mapping Solutions for providing the real-world MLS point clouds. We are also grateful to the Munich Data Science Institute (MDSI) and Dr. Ricardo Acevedo Cabra for his support through the TUM Data Innovation Lab. Finally, we thank the Data Innovation Lab members — Patrick Madlindl, Xinyuan Zhu, and Florian Hauck — for their valuable contributions.

This work is supported by the German Federal Ministry of Transport and Digital Infrastructure (BMVI) within the *Automated and Connected Driving* funding program under the Grant No. 01MM20012K (SAVeNoW).

References

[1] ASAM. OpenDRIVE – Format Specification, Rev. 1.4. https://www.asam.net/standards/detail/opendrive/, 2015. Accessed: 2024-02-15. 3, 4

[2] Authors. Mind the domain gap: Measuring the domain gap between real-world and synthetic point clouds for automated driving development. *Supplied as supplementary material: preprint_anonymized.pdf*, 2025. 2, 5

[3] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Juergen Gall. SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences. In *Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9297–9307, Seoul, South Korea, 2019. 6

[4] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A Multimodal Dataset for Autonomous Driving. In *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11618–11628, Virtual, 2020. 6

[5] Meida Chen, Qingyong Hu, Zifan Yu, Hugues Thomas, Andrew Feng, Yu Hou, Kyle McCullough, Fengbo Ren, and Lucio Soibelman. Stpls3d: A large-scale synthetic and real aerial photogrammetry 3d point cloud dataset. *arXiv preprint arXiv:2203.09065*, 2022. 2

[6] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-Annotated 3D Reconstructions of Indoor Scenes. In *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2432–2443, Honolulu, HI, USA, 2017. 4, 8

[7] Mark De Deuge, Alastair Quadros, Calvin Hung, and Bertrand Douillard. Unsupervised Feature Learning for Classification of Outdoor 3D Scans. *Australasian Conference on Robotics and Automation*, 2(1), 2013. 4

[8] Jean-Emmanuel Deschaud, David Duque, Jean Pierre Richa, Santiago Velasco-Forero, Beatriz Marcotegui, and François Goulette. Paris-CARLA-3D: A Real and Synthetic Outdoor Point Cloud Dataset for Challenging Tasks in 3D Mapping. *Remote Sensing*, 13(22), 2021. 1, 2, 4, 6

[9] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning (CoRL)*, pages 1–16, Mountain View, CA, USA, 2017. PMLR. 1, 2, 4

[10] Thomas Froech, Benedikt Schwab, and Olaf Wysocki. CityGML2OBJ 2.0: Command line converter of CityGML (.gml) to OBJ (.obj) files, while maintaining the semantics. https://github.com/tum-gis/CityGML2OBJv2, 2023. Accessed: 2023-08-23. 3

[11] Adrien Gaidon, Qiao Wang, Yohann Cabon, and Eleonora Vig. Virtual worlds as proxy for multi-object tracking analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4340–4349, 2016. 3

[12] Biao Gao, Yancheng Pan, Chengkun Li, Sibo Geng, and Huijing Zhao. Are We Hungry for 3D LiDAR Data for Semantic Segmentation? A Survey of Datasets and Methods. *IEEE Transactions on Intelligent Transportation Systems*, 23(7): 6063–6081, 2022. 2

[13] Jakob Geyer, Yohannes Kassahun, Mentar Mahmudi, Xavier Ricou, Rupesh Durgesh, Andrew S Chung, Lorenz Hauswald, Viet Hoang Pham, Maximilian Mühlegg, Sebastian Dorn, et al. A2D2: Audi autonomous driving dataset. *arXiv preprint arXiv:2004.06320*, 2020. 4

[14] Silvia María González-Collazo, Jesús Balado, Iván Garrido, Javier Grandío, Rabia Rashdi, Elisavet Tsiranidou, Pablo del Río-Barral, Erik Rúa, Iván Puente, and Henrique Lorenzo. Santiago urban dataset SUD: Combination of Handheld and Mobile Laser Scanning point clouds. *Expert Systems with Applications*, 238:121842, 2024. 4

[15] Silvia María González-Collazo, Benedikt Schwab, Christof Beil, Thomas H. Kolbe, Elena González, and Jesús Balado. Curbside management from MLS and HMLS point clouds to CityGML 3.0. *Geo-spatial Information Science*, 0(0):1–23, 2025. 3

[16] David Griffiths and Jan Boehm. SynthCity: A large scale synthetic point cloud. *arXiv preprint arXiv:1907.04758*, 2019. 4

[17] Gerhard Gröger, Thomas H Kolbe, Claus Nagel, and Karl-Heinz Häfele. OGC City Geography Markup Language CityGML Encoding Standard, 2012. Open Geospatial Consortium: Wayland, MA, USA, 2012. 3, 4

[18] Timo Hackel, Nikolay Savinov, Lubor Ladicky, Jan D Wegner, Konrad Schindler, and Marc Pollefeys. Semantic3D.net: A new Large-scale Point Cloud Classification Benchmark. *arXiv preprint arXiv:1704.03847*, 2017. 4

[19] Andreas Haigermoser, Bernd Luber, Jochen Rauh, and Gunnar Gräfe. Road and track irregularities: Measurement, assessment and simulation. *Vehicle System Dynamics*, 53(7): 878–957, 2015. 3

[20] Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Niki Trigoni, and Andrew Markham. Randla-net: Efficient semantic segmentation of large-scale point clouds. In *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, 2020. 7, 1, 2, 3

[21] Sebastian Huch, Luca Scalerandi, Esteban Rivera, and Markus Lienkamp. Quantifying the LiDAR Sim-to-Real Domain Shift: A Detailed Investigation Using Object Detectors and Analyzing Point Clouds at Target-Level. *IEEE Transactions on Intelligent Vehicles*, 2023. 1

[22] Lingdong Kong, Youquan Liu, Xin Li, Runnan Chen, Wenwei Zhang, Jiawei Ren, Liang Pan, Kai Chen, and Ziwei Liu. Robo3d: Towards robust and reliable 3d perception against corruptions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19994–20006, 2023. 2

[23] Loïc Landrieu and Martin Simonovsky. Large-scale point cloud semantic segmentation with superpoint graphs. In *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4558–4567, Salt Lake City, UT, USA, 2018. 1

[24] Chenfei Liao, Kaiyu Lei, Xu Zheng, Junha Moon, Zhixiong Wang, Yixuan Wang, Danda Pani Paudel, Luc Van Gool, and

Xuming Hu. Benchmarking multi-modal semantic segmentation under sensor failures: Missing and noisy modality robustness. In *Proceedings of the 2025 IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, pages 1576–1586, Nashville, TN, USA, 2025. 2, 4

[25] Yiyi Liao, Jun Xie, and Andreas Geiger. KITTI-360: A novel dataset and benchmarks for urban scene understanding in 2D and 3D. *arXiv preprint arXiv:2109.13410*, 2021. 4

[26] Yaping Lin, George Vosselman, and Michael Ying Yang. Weakly supervised semantic segmentation of airborne laser scanning point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 187:79–100, 2022. 2

[27] Romain Loiseau, Mathieu Aubry, and Loïc Landrieu. Online Segmentation of LiDAR Sequences: Dataset and Algorithm. In *Proceedings of the 2022 European Conference on Computer Vision (ECCV)*, pages 301–317, Cham, 2022. Springer Nature Switzerland. 4

[28] Francesca Matrone, Eleonora Grilli, Massimo Martini, Marina Paolanti, Roberto Pierdicca, and Fabio Remondino. Comparing machine and deep learning methods for large 3D heritage semantic segmentation. *ISPRS International Journal of Geo-Information*, 9(9):535, 2020. 4

[29] Francesca Matrone, Andrea Lingua, Roberto Pierdicca, Eva Savina Malinverni, Marina Paolanti, Eleonora Grilli, Fabio Remondino, Arnadi Murtiyoso, and Tania Landes. A Benchmark for Large-Scale Heritage Point Cloud Semantic Segmentation. *ISPRS International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLIII-B2-2020:1419–1426, 2020. 3

[30] Daniel Munoz, J. Andrew Bagnell, Nicolas Vandapel, and Martial Hebert. Contextual classification with functional Max-Margin Markov networks. In *Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 975 – 982, Miami, FL, USA, 2009. 4

[31] Sergey I Nikolenko et al. *Synthetic data for deep learning*. Springer, 2021. 3

[32] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 652–660, Honolulu, HI, USA, 2017. 7, 1, 3

[33] Charles R. Qi, Li Yi, Hao Su, and Leonidas J. Guibas. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS)*, page 5105–5114, Red Hook, NY, USA, 2017. Curran Associates Inc. 7, 1, 3

[34] Damien Robert, Hugo Raguet, and Loïc Landrieu. Efficient 3D semantic segmentation with superpoint transformer. In *Proceedings of the 2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, Paris, France, 2023. 7, 1, 2, 4

[35] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M Lopez. The Synthia Dataset: A Large Collection of Synthetic Images for Semantic Segmentation of Urban Scenes. In *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3234–3243, Las Vegas, NV, USA, 2016. 2

[36] Xavier Roynard, Jean-Emmanuel Deschaud, and François Goulette. Paris-Lille-3D: A large and high-quality ground-truth urban point cloud dataset for automatic segmentation and classification. *International Journal of Robotics Research*, 37(6):545–557, 2018. 4

[37] Benedikt Schwab and Thomas H. Kolbe. Requirement analysis of 3D road space models for automated driving. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, IV-4/W8:99–106, 2019. 3

[38] Benedikt Schwab and Thomas H. Kolbe. Validation of parametric OpenDRIVE road space models. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, X-4/W2-2022:257–264, 2022. 4

[39] Benedikt Schwab, Christof Beil, and Thomas H Kolbe. Spatio-Semantic Road Space Modeling for Vehicle–Pedestrian Simulation to Test Automated Driving Systems. *Sustainability*, 12(9):3799, 2020. 3, 4

[40] Andrés Serna, Beatriz Marcotegui, François Goulette, and Jean-Emmanuel Deschaud. Paris-rue-Madame database: As 3D mobile laser scanner dataset for benchmarking urban detection, segmentation and classification methods. In *Proceedings of the Third International Conference on Pattern Recognition Applications and Methods (ICPRAM)*, pages 819–824, Angers, France, 2014. 4

[41] Shital Shah, Debadeepta Dey, Chris Lovett, and Ashish Kapoor. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and Service Robotics*, pages 621–635, Cham, 2018. Springer International Publishing. 1, 2

[42] Shuo Shen, Yan Xia, Andreas Eich, Yusheng Xu, Bisheng Yang, and Uwe Stilla. SegTrans: Semantic Segmentation With Transfer Learning for MLS Point Clouds. *IEEE Geoscience and Remote Sensing Letters*, 20:1–5, 2023. 2

[43] 3D Mapping Solutions. https://www.3d-mapping.de/, 2025. Accessed: 2025-08-12. 3

[44] Steven Spiegel and Jorge Chen. Using Simulation Data from Gaming Environments for Training a Deep Learning Algorithm on 3D Point Clouds. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, VIII-4/W2-2021:67–74, 2021. 1, 4

[45] Andrea Stocco, Brian Pulfer, and Paolo Tonella. Mind the Gap! A Study on the Transferability of Virtual Versus Physical-World Testing of Autonomous Driving Systems. *IEEE Transactions on Software Engineering*, 49(4):1928–1940, 2023. 2

[46] Weikai Tan, Nannan Qin, Lingfei Ma, Ying Li, Jing Du, Guorong Cai, Ke Yang, and Jonathan Li. Toronto-3D: A Large-scale Mobile LiDAR Dataset for Semantic Segmentation of Urban Roadways. In *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 202–203, Los Alamitos, CA, USA, 2020. 4, 6

[47] Wenzhao Tang, Weihang Li, Xiucheng Liang, Olaf Wysocki, Filip Biljecki, Christoph Holst, and Boris Jutzi. Texture2lod3: Enabling lod3 building reconstruction with panoramic images. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 2016–2026, 2025. 1

[48] Hugues Thomas, Charles R. Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas Guibas. KPConv: Flexible and Deformable Convolution for Point Clouds. In *Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6410–6419, Seoul, South Korea, 2019. 7, 1, 2, 3

[49] Bruno Vallet, Mathieu Brédif, Andrés Serna, Beatriz Marcotegui, and Nicolas Paparoditis. TerraMobilita/iQmulus urban point cloud analysis benchmark. *Computers & Graphics*, 49:126–133, 2015. 4

[50] Jingyi Wang, Yu Liu, Hanlin Tan, and Maojun Zhang. A survey on weakly supervised 3d point cloud semantic segmentation. *IET Computer Vision*, 18(3):329–342, 2024. 2

[51] Peng-Shuai Wang. Octformer: Octree-based transformers for 3D point clouds. *ACM Transactions on Graphics (SIGGRAPH)*, 42(4), 2023. 7, 1, 2, 4

[52] Qingwang Wang, Mingye Wang, Jiangbo Huang, Tianzhu Liu, Tao Shen, and Yanfeng Gu. Unsupervised domain adaptation for cross-scene multispectral point cloud classification. *IEEE Transactions on Geoscience and Remote Sensing*, 62:1–15, 2024. 2

[53] Lukas Winiwarter, Alberto Manuel Esmorís Pena, Hannah Weiser, Katharina Anders, Jorge Martínez Sánchez, Mark Searle, and Bernhard Höfle. Virtual Laser Scanning with HELIOS++: A Novel Take on Ray Tracing-Based Simulation of Topographic Full-Waveform 3D Laser Scanning. *Remote Sensing of Environment*, 269:112772, 2022. 2

[54] Chengzhi Wu, Xuelei Bi, Julius Pfrommer, Alexander Cebulla, Simon Mangold, and Jürgen Beyerer. Sim2real transfer learning for point cloud segmentation: An industrial application case on autonomous disassembly. In *Proceedings of the 2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 4520–4529, Waikoloa, HI, USA, 2023. 2

[55] Xiaoyang Wu, Li Jiang, Peng-Shuai Wang, Zhijian Liu, Xihui Liu, Yu Qiao, Wanli Ouyang, Tong He, and Hengshuang Zhao. Point Transformer V3: Simpler, Faster, Stronger. In *Proceedings of the 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4840–4851, 2024. 7, 1, 2, 4

[56] Olaf Wysocki, Ludwig Hoegner, and Uwe Stilla. TUM-FAÇADE: Reviewing and enriching point cloud benchmarks for façade segmentation. *ISPRS International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLVI-2/W1-2022:529–536, 2022. 3, 4

[57] Olaf Wysocki, Yan Xia, Magdalena Wysocki, Eleonora Grilli, Ludwig Hoegner, Daniel Cremers, and Uwe Stilla. Scan2LoD3: Reconstructing Semantic 3D Building Models at LoD3 Using Ray Casting and Bayesian Networks. In *Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 6547–6557, Vancouver, BC, Canada, 2023. 2, 3

[58] Olaf Wysocki, Benedikt Schwab, Christof Beil, Christoph Holst, and Thomas H. Kolbe. Reviewing open data semantic 3d city models to develop novel 3d reconstruction methods. *ISPRS International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLVIII-4-2024:493–500, 2024. 3

[59] Olaf Wysocki, Yue Tan, Thomas Froech, Yan Xia, Magdalena Wysocki, Ludwig Hoegner, Daniel Cremers, and Christoph Holst. Zaha: Introducing the level of facade generalization and the large-scale point cloud facade semantic segmentation benchmark dataset. In *Proceedings of the 2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 7648–7658. IEEE, 2025. 2, 3, 4

[60] Aoran Xiao, Jiaxing Huang, Dayan Guan, Fangneng Zhan, and Shijian Lu. Transfer Learning from Synthetic to Real LiDAR Point Cloud for Semantic Segmentation. In *Proceedings of the 2022 Association for the Advancement of Artificial Intelligence Conference on Artificial Intelligence (AAAI)*, pages 2795–2803, Arlington, VA, USA, 2022. 2

[61] Changyu Zeng, Wei Wang, Anh Nguyen, Jimin Xiao, and Yutao Yue. Self-supervised learning for point cloud data: A survey. *Expert Systems with Applications*, 237:121354, 2024. 2

[62] Wuming Zhang, Jianbo Qi, Peng Wan, Hongtao Wang, Donghui Xie, Xiaoyan Wang, and Guangjian Yan. An Easy-to-Use Airborne LiDAR Data Filtering Method Based on Cloth Simulation. *Remote Sensing*, 8:501, 2016. 3

[63] Zaiwei Zhang, Rohit Girdhar, Armand Joulin, and Ishan Misra. Self-Supervised Pretraining of 3D Features on any Point-Cloud. In *Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10252–10263, Montreal, QC, Canada, 2021. 2

[64] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip Torr, and Vladlen Koltun. Point Transformer. In *Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 16239–16248, Montreal, QC, Canada, 2021. 7, 1, 2, 4

[65] Jingwei Zhu, Joachim Gehrung, Rong Huang, Björn Borgmann, Zhenghao Sun, Ludwig Hoegner, Marcus Hebel, Yusheng Xu, and Uwe Stilla. TUM-MLS-2016: An annotated mobile LiDAR dataset of the TUM City Campus for semantic point cloud interpretation in urban areas. *Remote Sensing*, 12(11):1875, 2020. 4

TrueCity: Real and Simulated Urban Data for Cross-Domain 3D Scene Understanding

Supplementary Material

7. Laser Scanning Simulation

7.1. Configuration

For the laser scanning simulation, we configure the sensor to match the real-world laser scanning setup.

Parameter	Value
Number of lasers	128
Points generated by all lasers	500,000 points/s
Rotation frequency	20 Hz
Upper FOV	15°
Lower FOV	−25°
Horizontal FOV	360°
Range	100 m

Table 5. Configuration parameters of the LiDAR sensor model.

7.2. Radiometry

As highlighted in the main body of the paper, we do not include any radiometric features in the simulation (e.g., RGB). We opt for this approach due to the limited field-of-view of the acquisition camera, which does not fully cover the field-of-view of the laser scanners. This issue would result in incomplete and non-robust image-to-model projections. A possible solution for model texturing is presented in [47]; however, it is subject to wide-angle facade observation, and the choice of the appropriate image for texturing is based on a heuristic, which introduces further randomness to the scan-to-model domain gap analysis. Nevertheless, we are convinced that future work shall further investigate the impact of radiometry and thus object material on the simulation and resulting domain gap.

7.3. Dynamic Objects

The dynamic objects are not included in the simulation setup, as discussed in the main body of the paper. The large discrepancy between the accurate as-it-happened simulation and actual measurements dictates this choice. Even if the procedural vehicle and pedestrian models can be placed in a scene, the simulation of their realistic trajectories is questionable and would include large heuristics in the simulation, which would make the domain gap analysis hardly tractable. Additionally, due to the speed of the mapping vehicle and the observed vehicles and pedestrians, the dynamic object representation is sparse and mainly noisy. We observe the impact of this phenomenon in Tab. 4, where the segmentation results on real-only data also show a large discrepancy between the baseline models.

8. Experimental Setup

8.1. Baseline Segmentation Models

Point-based models operate directly on unordered points with permutation-invariant set functions and lightweight neighborhood aggregation. In our experiments, we use PointNet [32], PointNet++ [33], and RandLA-Net [20]. PointNet applies per-point MLPs with a symmetric reduction for global context; PointNet++ adds hierarchical sampling and grouping to capture local structure; RandLA-Net improves scalability via random sampling with attentive local aggregation.

Kernel-based models 3D convolutions impose locality and yield predictable receptive-field growth. We use KPConv [48], which places learnable kernel points within each neighborhood and aggregates features with geometry-aligned weights in an encoder-decoder pyramid, preserving small-scale structure while expanding context.

Transformer-based models attention replaces fixed kernels with content-adaptive routing across local neighborhoods and long-range, semantically related regions. We consider region-level attention with Superpoint Transformer [34] and octree-structured attention with OctFormer [51], as well as point-level attention with Point Transformer v1 and v3 [55, 64]. Superpoint Transformer attends over a superpoint graph [23]; OctFormer organizes tokens on an octree for multi-scale attention; Point Transformer uses localized self-attention with relative positional encodings, and v3 tightens geometric invariances and reduces overhead for large-scale outdoor LiDAR.

8.2. Training Setup

Unless otherwise noted, we maintain a unified training setup: all models are trained for 100 epochs with a constant learning rate of 10^{-4}, a mini-batch size of 32, and the AdamW optimizer. Model-specific deviations from this setup are described in Sec. 9. Experiments run on NVIDIA H40, L40, and RTX 6000 Ada Generation GPUs; to isolate the effect of the real/synthetic composition, we keep the training dynamics (optimizer, schedule, batch size, total epochs, augmentations, and point budget) identical across conditions and architectures. Whenever possible, we

rely on the authors' public implementations with minimal changes.

8.3. Evaluation Setup

Evaluation follows the same point budget and normalization as training. We report three standard metrics for semantic segmentation: mean Intersection-over-Union (mIoU), mean Accuracy (mAcc), and Overall Accuracy (OA). For C semantic classes, let n_{ij} denote the number of points of class i predicted as class j, and $n_i = \sum_j n_{ij}$ the total number of points in class i. The metrics are defined as:

$$\text{IoU}_i = \frac{n_{ii}}{n_i + \sum_j n_{ji} - n_{ii}}, \tag{2}$$

$$\text{mIoU} = \frac{1}{C} \sum_{i=1}^{C} \text{IoU}_i, \tag{3}$$

$$\text{Acc}_i = \frac{n_{ii}}{n_i}, \tag{4}$$

$$\text{mAcc} = \frac{1}{C} \sum_{i=1}^{C} \text{Acc}_i, \tag{5}$$

$$\text{OA} = \frac{\sum_{i=1}^{C} n_{ii}}{\sum_{i=1}^{C} n_i}. \tag{6}$$

9. Training Recipe

9.1. PointNet and PointNet++

Each input point cloud is downsampled to 2,048 points via farthest-point sampling to preserve geometric coverage. We apply a fixed augmentation regimen: normalization to the unit sphere, a random rotation about the vertical (z) axis, independent reflections across the x and y axes, isotropic scaling drawn from a uniform distribution, and additive jitter. The networks are trained with cross-entropy under the global settings of Sec. 8.

9.2. RandLA-Net

Training operates on pre-sharded point clouds. For each iteration we form a mini-batch by uniformly sampling a fixed-size subset of 2,048 points per sample; at evaluation time we sweep scenes with sequential, non-overlapping chunks. Augmentations follow the original recipe of [20] (normalization, random z-rotation, axis flips, isotropic scaling, elastic distortion, and jitter). Optimization and schedules are identical to Sec. 8.

9.3. KPConv

We use the reference KPConv segmentation model [48] with the authors' architectural defaults. To ensure parity across methods, each training item is reduced to 2,048 points via farthest-point sampling and processed with the same augmentation regimen as PointNet/PointNet++. The model is optimized with AdamW at a constant learning rate of 10^{-4}; loss weighting follows the effective-number scheme.

9.4. Point Transformer v1 and v3

For Point Transformer v1 [64] and v3 [55], inputs are standardized to 2,048 points per item using farthest-point sampling. We apply the same normalization and geometric augmentations as above to avoid confounding from data processing. Both variants are trained for 100 epochs with AdamW (10^{-4} learning rate) and cross-entropy with class weighting as in Sec. 8. Evaluation uses the identical point budget and normalization.

9.5. OctFormer

We follow the reference implementation of OctFormer [51]. Coordinates are normalized to the unit sphere and encoded into an OCNN octree (depth 8, full_depth 2) with the associated neighborhood structures. The OctFormer backbone feeds an FPN-style head that upsamples multi-scale features and interpolates them to query points for per-point classification. Unless stated otherwise, we sample 2,048 points per item for parity with the other models and train with AdamW at 10^{-4}, cross-entropy (with `ignore_label`, optional class weights and label smoothing), and the global settings of Sec. 8.

9.6. Superpoint Transformer

We adopt the Superpoint Transformer (SPT) [34] in its reference implementation. As per the SPT methodology, each input scene is oversegmented into superpoints, so no further downsampling or chunking is required. Preprocessing and augmentation follow the DALES configuration in the official codebase. To remain faithful to the original setup, training uses stochastic gradient descent (SGD) with a learning rate of 0.01, weight decay of 10^{-4}, and a batch size of 4, which differs slightly from the global setting in Sec. 8.

9.7. Data Augmentation

We acknowledge that the data augmentation is a common technique for improving model performance and increasing its generalization [22]. However, we opted to omit data augmentation in our experimental setup, since it would have introduced an additional variable to the coherent domain gap analysis: The influence of data augmentation would make assessing the impact of real-to-synthetic data hardly tractable.

10. Additional Information about Dataset and Evaluation Result

Class / Metric	PointNet					PointNet++				
	100S	75S	50S	25S	0S	100S	75S	50S	25S	0S
mIoU	6.03	10.74	10.89	13.10	**14.51**	9.72	20.95	23.18	**25.38**	23.39
mAcc	14.58	21.63	21.82	25.35	**25.98**	19.75	29.37	35.22	30.84	**38.03**
OA	30.36	48.10	49.29	47.99	**49.82**	34.39	62.80	**65.36**	63.27	63.15
RoadSurface	7.84	51.92	54.04	49.78	**67.55**	60.90	73.90	**81.20**	80.30	72.50
GroundSurface	16.45	24.11	17.27	21.90	**28.55**	33.20	45.30	**50.40**	49.60	32.40
CityFurniture	0.62	2.68	4.12	**10.80**	8.56	15.20	14.60	21.30	25.50	**46.20**
Vehicle	0.00	0.00	0.00	0.00	0.00	**3.50**	0.00	0.18	0.51	0.01
Pedestrian	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**0.32**	0.04
WallSurface	39.97	37.01	**45.28**	43.07	30.87	0.20	56.90	63.50	**65.70**	63.20
RoofSurface	0.00	0.06	**0.19**	0.04	0.00	0.00	**0.80**	0.30	0.10	0.30
Door	0.02	0.00	0.00	**0.11**	0.00	0.00	0.00	0.20	0.10	**0.70**
Window	2.57	6.00	1.05	5.38	**6.39**	3.61	7.30	4.50	4.20	**14.60**
BuildingInstallation	0.01	0.26	0.01	**4.53**	0.15	0.00	0.70	3.00	1.70	**4.60**
SolitaryVegetationObject	4.24	1.04	5.78	11.40	**14.79**	0.00	**50.60**	39.30	45.60	14.40
Noise	0.63	5.82	2.93	10.21	**17.26**	0.00	1.30	14.30	30.90	**31.70**

Table 6. Per-class IoU (↑) under different S–R dataset mixtures. Models: **PointNet** [32] and **PointNet++** [33]. Bold marks the best value for each model across mixtures.

Class / Metric	RandLA-Net					KPConv				
	100S	75S	50S	25S	0S	100S	75S	50S	25S	0S
mIoU	8.98	13.25	15.73	16.89	**17.71**	15.84	21.55	28.50	22.33	**29.90**
mAcc	25.62	**31.27**	28.48	29.82	30.97	38.12	33.59	37.79	**42.91**	42.01
OA	35.40	50.32	**59.37**	57.09	54.57	50.07	62.08	61.62	61.92	**62.80**
RoadSurface	3.12	60.53	**70.35**	62.01	60.26	30.64	**69.26**	54.26	56.31	54.59
GroundSurface	23.31	**29.51**	29.41	25.27	26.65	24.67	**36.49**	31.91	33.20	35.96
CityFurniture	2.33	**15.91**	9.99	14.38	12.93	**18.25**	10.09	12.67	1.85	14.57
Vehicle	0.00	0.03	1.03	0.59	**3.27**	0.00	0.00	38.93	**70.33**	63.64
Pedestrian	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
WallSurface	45.19	41.29	51.15	**54.18**	53.23	59.97	63.21	68.54	**73.55**	71.03
RoofSurface	**0.32**	0.17	0.29	0.19	0.18	0.82	0.01	0.47	0.00	**1.79**
Door	**18.22**	2.51	1.41	0.16	3.25	0.00	0.00	0.00	0.00	0.00
Window	3.82	2.83	9.51	8.75	**9.72**	1.86	0.43	**18.21**	1.17	2.89
BuildingInstallation	0.94	1.68	2.91	**8.46**	2.90	**7.85**	3.02	0.85	0.85	3.72
SolitaryVegetationObject	10.42	1.91	2.07	10.68	**14.86**	45.74	70.51	**81.45**	0.41	77.13
Noise	0.10	2.64	10.63	17.95	**25.27**	0.31	5.58	**34.73**	30.26	33.43

Table 7. Per-class IoU (↑) under different S–R dataset mixtures. Models: **RandLA-Net** [20] and **KPConv** [48]. Bold marks the best value for each model across mixtures.

Class / Metric	Point Transformer v1					Point Transformer v3				
	100S	75S	50S	25S	0S	100S	75S	50S	25S	0S
mIoU	16.30	19.79	23.43	24.66	**28.89**	14.13	19.29	**25.30**	24.64	25.24
mAcc	26.09	34.39	37.01	35.52	**39.33**	28.07	34.09	43.16	40.33	**44.09**
OA	57.54	60.29	67.54	**68.70**	67.98	53.15	60.22	**65.94**	65.72	60.75
RoadSurface	62.41	61.89	**80.33**	77.93	70.44	64.42	68.85	70.74	**71.17**	42.36
GroundSurface	31.94	37.54	**45.54**	37.09	41.67	32.53	32.66	**33.27**	32.80	27.53
CityFurniture	25.89	19.50	13.51	14.23	**44.83**	5.35	34.86	24.11	17.34	**35.13**
Vehicle	0.00	0.51	4.41	1.51	**6.54**	0.00	0.00	0.00	0.00	0.00
Pedestrian	0.00	0.00	**0.05**	0.00	0.00	0.00	0.00	0.00	0.00	0.00
WallSurface	53.15	61.42	64.39	65.32	**67.10**	48.15	54.79	63.42	64.63	**69.69**
RoofSurface	0.10	0.29	**0.92**	0.30	0.08	0.01	0.10	0.10	0.00	**0.36**
Door	0.01	0.13	0.51	**0.66**	0.01	0.00	0.81	**2.77**	0.08	1.62
Window	3.28	5.61	7.61	11.78	**31.29**	3.09	5.17	4.92	4.35	**10.70**
BuildingInstallation	3.08	5.24	6.69	**8.87**	3.41	8.40	2.53	**9.95**	9.30	4.44
SolitaryVegetationObject	15.54	35.35	34.70	43.22	**52.31**	6.72	12.90	60.86	60.80	**70.39**
Noise	0.16	9.98	22.43	**35.04**	28.96	0.84	18.80	33.50	35.19	**40.69**

Table 8. Per-class IoU (↑) under different S–R dataset mixtures. Models: **Point Transformer v1** [64] and **Point Transformer v3** [55]. Bold marks the best value for each model across mixtures.

Class / Metric	OctFormer					Superpoint Transformer				
	100S	75S	50S	25S	0S	100S	75S	50S	25S	0S
mIoU	13.07	14.17	14.22	13.91	**17.65**	14.31	17.01	14.22	**19**.61	15.96
mAcc	22.84	23.99	26.28	26.15	**27.19**	24.28	29.40	26.42	**31.79**	28.29
OA	53.30	55.34	49.71	50.97	**56.28**	54.17	**58.62**	54.63	56.98	54.64
RoadSurface	49.32	51.58	35.34	54.81	**71.98**	**77.74**	73.73	61.19	63.51	53.16
GroundSurface	16.01	16.34	**22.60**	19.60	16.99	18.09	30.35	16.83	**31.35**	25.92
CityFurniture	0.92	2.18	**15.24**	0.01	0.14	1.22	**3.70**	0.85	2.49	0.47
Vehicle	0.00	0.00	0.00	0.07	**20.82**	0.00	0.26	0.55	**4.82**	0.22
Pedestrian	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
WallSurface	56.22	**62.14**	60.36	45.79	38.96	38.97	47.32	55.57	56.98	**63.19**
RoofSurface	0.00	0.00	0.00	0.00	0.00	2.11	2.60	2.84	**3.21**	1.25
Door	0.00	0.00	0.00	0.00	0.00	**0.30**	0.09	0.00	0.00	0.00
Window	0.00	0.00	**0.76**	0.00	0.00	**8.94**	3.89	1.43	2.99	0.55
BuildingInstallation	0.00	0.00	**4.02**	0.00	0.00	2.29	0.75	0.01	1.30	**2.63**
SolitaryVegetationObject	34.33	37.85	32.25	26.72	**47.30**	21.54	34.45	16.70	**48.13**	21.12
Noise	0.00	0.00	0.08	**19.93**	15.57	0.46	6.96	14.70	20.50	**23.07**

Table 9. Per-class IoU (↑) under different S–R dataset mixtures. Models: **OctFormer** [51] and **Superpoint Transformer** [34]. Bold marks the best value for each model across mixtures.

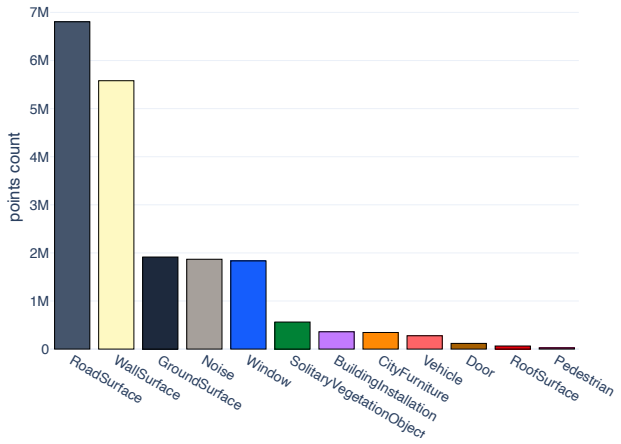

Figure 6. The class distribution of the real-only test set.

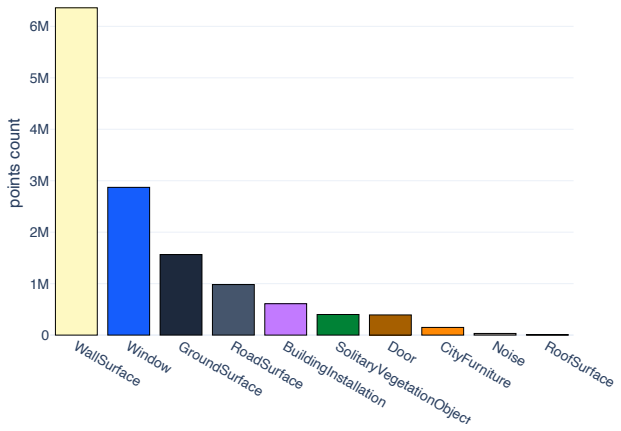

Figure 7. The class distribution of the synthetic-only test set, which is not used in training and evaluation.

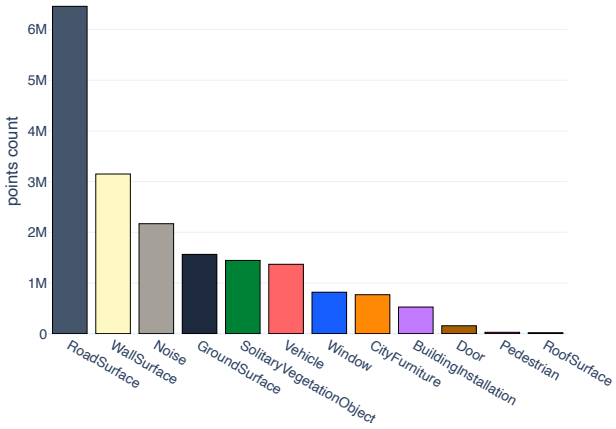

Figure 8. The class distribution of the real-only validation set.

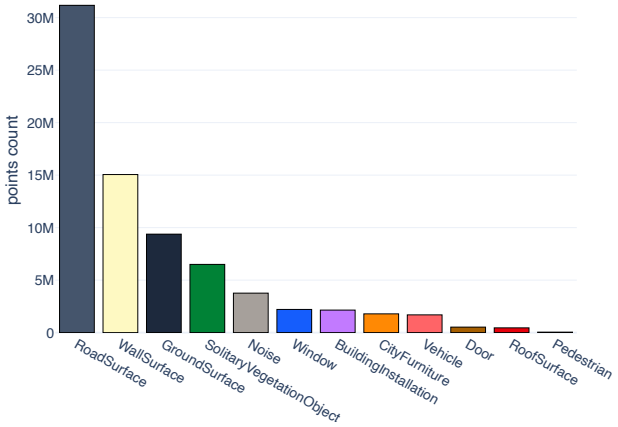

Figure 9. The class distribution of the 0S–100R train set.

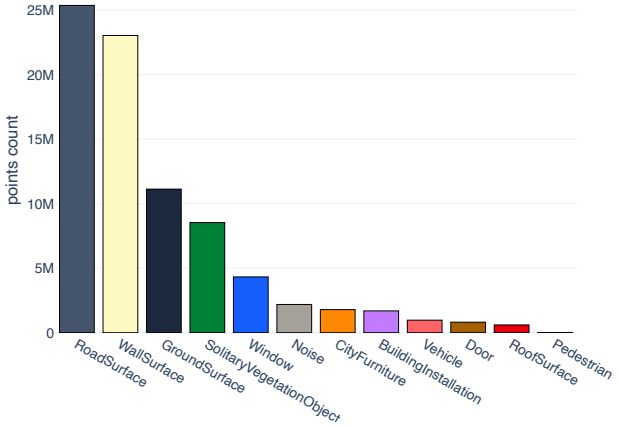

Figure 10. The class distribution of the 25S–75R train set.

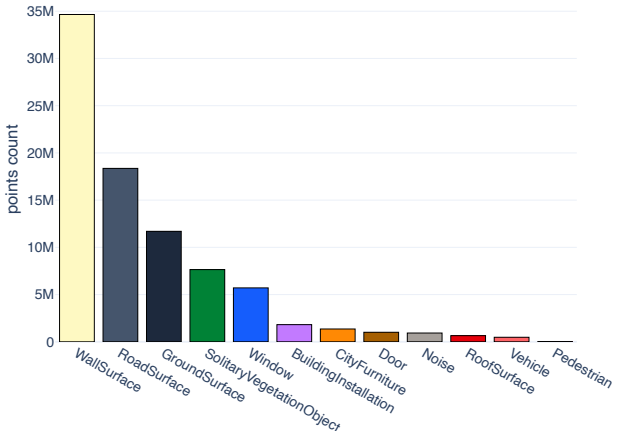

Figure 11. The class distribution of the 50S–50R train set.

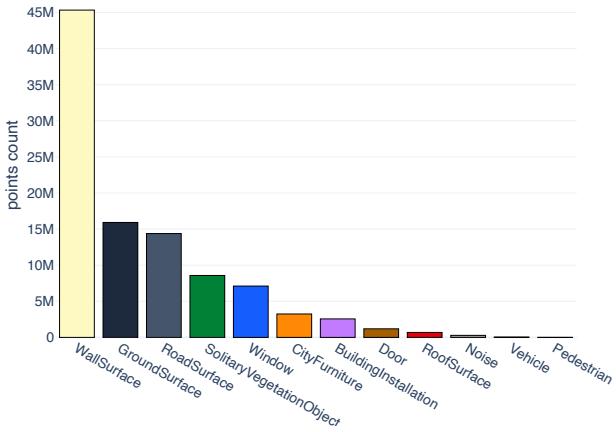

Figure 12. The class distribution of the 75S–25R train set.

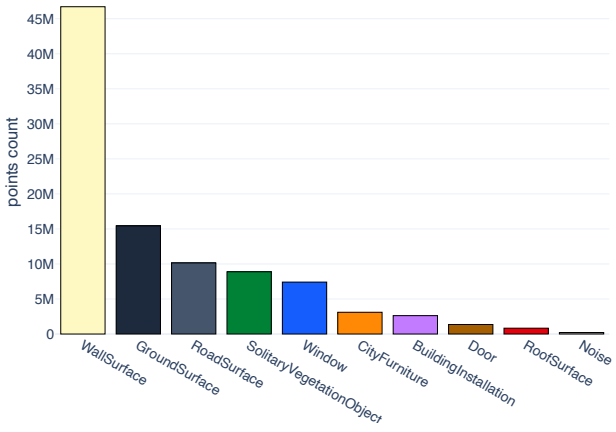

Figure 13. The class distribution of the 100S–0R train set.